

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:)
Reiner Kraft, et al.) Group Art Unit: 2157
Serial No.: 09/607,289)
Confirmation No. 8226)
Filed: June 30, 2000) Examiner: Sahera HALIM
For: SYSTEM FOR MANAGING)
AN EXCHANGE OF QUESTIONS)
AND ANSWERS THROUGH AN)
EXPERT ANSWER WEB SITE)

37 C.F.R. 1.131 AFFIDAVIT

We, the undersigned, the inventors of the above-referenced patent application, hereby declare the following:

- 1) The pending claims of our above identified patent invention were rejected under 35 U.S.C. 102(e) and 35 U.S.C. §103(a) based on the prior art reference of Stephanou (U.S. 6,505,166) with a effective filing date of November 23, 1999 based on Provisional Application Number 60/167,065 and issued on January 7, 2003 (hereinafter referred to as "Stephanou").
- 2) The invention described in the above referenced patent application was reduced to a writing and signed by the undersigned applicants prior to the November 23, 1999 date of Stephanou. In particular, the relevant portion of our Invention Disclosure upon which the above referenced patent application was based is attached herewith.

We, the undersigned, declare all of the above statements are made on our own knowledge, the above statements are true and correct, and the above statements are made on information that we believe to be true. We understand that false statements or concealment in obtaining a patent will subject us to fine and/or imprisonment or both (18 U.S.C. §1001) and may jeopardize the validity of the above identified patent application or any application issuing therefrom.



Reiner Kraft


Joann Ruvolo

Date: 10/2/03

Date: 10/2/03

BEST AVAILABLE COPY

	Disclosure ARC8-1999-0339
	Created By: Reiner Kraft Created On:
	Last Modified By: Reiner Kraft Last Modified On:
	*** IBM Confidential ***

Required fields are marked with the asterisk (*) and must be filled in to complete the form.

Summary

Status	Under Evaluation
Processing Location	ARC
Functional Area	DPB - Computer Science - (A.K. Chandra)
Attorney/Patent Professional	Allison Mortinger/Fishkill/Contr/IBM
IDT Team	Allison Mortinger/Fishkill/Contr/IBM; Sheila Hill/Almaden/IBM
Submitted Date	
Owning Division	RES
PVT Score	56
Lab	
Technology Code	
Incentive Program	

Inventors with Lotus Notes IDs

Inventors: Reiner Kraft/Almaden/IBM, Joann Ruvolo/Almaden/IBM

Inventor Name > denotes primary contact	Inventor Serial	Div/Dept	Manager Serial	Manager Name
Reiner Kraft	243800	22/RES	953543	Sundaresan, Neel/Kanar, Neel
Joann Ruvolo	243801	22/RES	953543	Sundaresan, Neel/Kanar, Neel

Inventors without Lotus Notes IDs

IDT Selection

IDT Team	Attorney/Patent Professional
Allison Mortinger/Fishkill/Contr/IBM	Allison Mortinger/Fishkill/Contr/IBM
Sheila Hill/Almaden/IBM	Sheila Hill/Almaden/IBM
Response Due to IDT	

Main Idea

Title of disclosure (in English):
System and Method for automatically managing response times in expertise systems based on an expert notification mechanism and dynamic ranking on responsiveness

1. Describe your invention, stating the problem solved (if appropriate), and indicating the advantages of using the invention.

The invention is related in the area of eCommerce and Knowledge management. Along with the Internet also rose the notion of a *virtual online community*. Popular web sites are using member

System and Method for automatically managing response times in expertise systems based on an expert notification mechanism and - c

databases and directory services. People can subscribe and register with such a site and join an online community. The benefit for users is to receive additional information, notification for events, online education, and other services. Also they are able to contribute with content, and share experience and knowledge with other members of this community. Because of the provided directory service, members of the community are able to easily communicate (e.g. email, chat, etc.) to share information and their expertise.

There are also online communities, which exists without having a registration database to keep track of their members. These are loosely organized communities, who share a common goal. A good example is the Open Source movement, which works together on the software development of free software (e.g. GNU, Linux, etc.). Because there doesn't exist a central database of their members, nor any directory service, it's more difficult to communicate with members, find specific skills and expertise sets.

Today more and more web services address the problem by providing an "Expertise Repository". For instance web sites like *AskExpert* (<http://www.askexpert.com>) provide a service for users to exchange information and expertise. These sites typically require a user to establish an user account with them. Then users have to create an expert profile. On the other side there are users who are seeking expertise. They can also register and post their questions. The next time an expert visits the site he/she might find the question of this user, while browsing the site. In addition, this expert might be able to provide a sufficient answer for the particular problem. Mostly, these systems also provide a mechanism that users are able to grade the answers of experts, to ensure quality and reward good answers.

These systems are already a step forwards in helping to promote information and knowledge exchange. However, they also suffer from problems, which we describe below. Also we provide a novel solution to solve these problems.

The first section will provide the following information:

1. Description of problem solved
2. Overview of the invention
3. Benefits and Advantages

1) Description of the problem solved

Today there are two identified severe problems in these concepts and implementations of exchanging expertise and knowledge:

1. There is no time constraint for a response

Typically someone poses a question in order to get a fast/quick response. It might be the case that someone is working on a project with a deadline. A problem needs to be solved prior to the deadline. The solution might be worthless afterwards. In current expertise repositories there's no real time constraint, which means that a person who poses a questions does not have the guarantee to receive a satisfactory answer within a specific time frame or receive an answer at all.

2. A potential expert might overlook an appropriate question

The current systems rely on pull mechanisms. Essentially an expert has to look what sort of questions are available. These questions are mostly sorted by category. An expert now has to manually browse through these categories, skim the headlines, and select an appropriate question. The chance that a question is overlooked, which would be easily answered by the expert, is very high. It would be desirable to have a push mechanism here, which would

System and Method for automatically managing response times in expertise systems based on an expert notification mechanism and - cc

automatically

- a) associate a question to a set of appropriate experts
- b) notify these experts about the question

We will describe below how the invention will address and solve these problems below.

2) Overview of the invention

Essentially the invention provides a solution for the problems described above. In particular the invention will

1. Associate a question/problem to a list of potential experts
2. Notify those experts about the new question/problem, which needs to be solved
3. Associate a timer with each notification
4. Manage responses from experts to ensure a question/problem is answered (correctly) within a specific time frame
5. Perform a dynamical ranking of experts within an expertise category based on response time

Thus the system can be easily integrated in existing web based expertise repositories. It will help to guarantee fast response times.

The idea is that once a potential list of experts is selected for a particular questions/problem, a sub set (window) of these potential experts will receive instantly a notification (e.g. email, pager, etc.). With each notification there's a time limit associated. An expert might be able to decline a question/problem, if it was not routed appropriately, or doesn't reflect the background knowledge of this expert. The invention will then make adjustments for future similar questions/problems to not include an expert, who has previously rejected a specific type (category) of questions/problems.

If an expert chooses to accept to solve a problem/question, he/she may do so, and send an email message (or fill out a web based form, etc.).

The timer for this notification is then stopped. Per default the system will wait until at least a minimum number of replies (threshold can be defined) are received for the problem/question. Then it will notify instantly the originator of the problem/question to gather feedback. Specifically the system will wait for the grading of the originator. In case a sufficient answer for the problem/question was received, the system will mark the process as successful, send feedback/rewards to the expert(s), who solved the problem, and stop the whole process. Especially the responsiveness ranking for each of these experts will increase, making those experts preferred for future problem solving processes. Note that the time the originator needs to perform the grading will not affect the experts responsiveness ranking. Time will only be measured from the moment the question/problem is raised, until the experts accepts the task, and eventually replies with a solution.

In case however that a set of experts where notified, and the associated timers expired without any response from these notified experts, the system will move the window to the next group of experts within the set (sliding window principle), until each potential expert was considered to solve the problem/question. Those experts, who didn't reply within the specified time frame will decrease in their responsiveness ranking. This means that someone who has a slow response time will gradually lose his "expert" status within the system, and so not considered early for new problems to tackle in future problem sessions. The key here is that there could be two persons (experts), who have the same knowledge. The expert with the lower responsiveness however will be considered for a later window in the problem solving process. Thus early (and correct) responses will be awarded. The goal is to provide a ranking for experts not only on their knowledge, but also on their time for responses. Of course it could also be the case, that a problem is not solvable at all. Or a problem is

System and Method for automatically managing response times in expertise systems based on an expert notification mechanism and - co

too complex to solve within a specific time. Experts have the ability to mark a problem as too complex to solve in the requested time.

The described process can be handled fully automatically by the proposed system, making it scalable. The system makes sure that all known experts for a particular problem/question will be notified. It will perform a dynamical ranking of experts based within a knowledge category based on response time, and correctness of answers. This will continuously optimize the process of selecting appropriate experts.

3) Benefits and advantages

There are two major advantages: First, every potential expert will be considered to solve a problem. Of course, experts with a higher responsiveness rating will be preferred. Because this is automatically done by the invention, potential experts will not be forgotten.

Second, response times are critical within an expertise repository. These will be tightly integrated into the problem solving process. Based on this dynamical ranking of experts the system will ensure optimized response times. Based on prior response times the system is able to provide an estimation for the time a problem solving process can take.

Overall we believe that the invention presents a novel idea of enhancing response times and quality within an expertise repository. We are not aware of an existing system, which provides such functionality. Thus the invention needs to be protected.

2. How does the invention solve the problem or achieve an advantage, (a description of "the invention", including figures inline as appropriate)?

The system comprises of the following components:

1. User/Session Manager
2. Expert Set Determinator
3. Sliding Window Manager
4. Expert Timer Manager
5. Response Manager
6. Ranking Processor
7. Expertise Repository (provided from Expertise Repository service provider)
8. Expertise Ranking Database (used by the invention for storing ranking information)

1. User/Session Manager

This component receives a new problem/question from a user entered using an electronic web based form for instance. The basic functionality of the User/Session Manager is typically embedded within a web application server. In addition, the User/Session Manager will query the Expertise Repository for the user's profile. Each question/problem is associated to a category. This information is needed to determine a list of available experts (see below). Overall this component has to make sure that

1. Only registered users of the Expertise Repository can initiate a problem solving request
2. Each problem has to be associated to a valid Expertise Repository category
3. Additional information, such as timing constraints, urgency etc. can be provided by the user. The User/Session Manager has to validate these.

After all the necessary data is collected, the User/Session Manager will pass this data to the

System and Method for automatically managing response times in expertise systems based on an expert notification mechanism and - cc

Expert Set Determinator for further processing. A message will be sent (email, pager, etc.) to the originator that the problem solving process was successfully initiated (optional).

2. Expert Set Determinator

The Expert Set Determinator will receive a problem solving request from the User/Session Manager. Each problem is associated with a category. Using this category information this component will query first the Expertise Repository, in order to retrieve a list of experts registered for this topic. An expert, once registered with the Expertise Repository, can register for a list of expertise categories, where he/she thinks to hold expertise in. The list of experts will be synchronized with the Expertise Database. Basically the Expertise Ranking Database will contain pointers into the Expertise Repository. Each expert will have an initial neutral ranking (e.g. 1.0). The ranking information, which is stored in the Expertise Ranking Database, will later on change based on the responsiveness, and the quality of the answers.

Essentially the Expert Set Determinator will create a list of all available experts, and sort them in decreasing order based on the ranking retrieved from the Expertise Ranking Database. This sorted/ranked list will then be forwarded to Sliding Window Manager.

3. Sliding Window Manager

This unit will now subsequently iterate through the process of problem solving, until

1. a specific amount of responses is received
2. the originator declares the problem as solved
3. all available experts are processed

First, we define the term "desired window size" = "w". A desired window size "w" represents a fixed number of experts (e.g. 10), for which the system will concurrently work in order to receive solutions within the problem solving process. At the beginning, the first w experts received from Expert Set Determinator, will be selected for the current active window. This list then will be passed to the Expert Timer Manager for processing.

After a while, the Sliding Window Manager will receive a notification from the Expert Timer Manager. There could be the following cases:

1. An expert accepted a problem/question for solving
2. An expert declined a problem/question
3. An expert canceled the current work on a problem/question
4. A timeout occurred

In each of the cases the new actual window size will be w-1. At this time the Sliding Window Manager has to provide the Response Manager with the name of an additional expert for the problem solving process. This will be the one with the next lowest ranking. The idea is that the Sliding Window Manager ensures, that the actual window size always equals to w (the desired window size). Of course, at some time the list of available experts will be empty, and so the actual window size will be 0, after all experts are processed.

It might also be the case, that there are a specific amount of responses available, which were received from the Response Manager. In this case these responses will be forwarded to the originator of the problem/question. The Sliding Window Manager will pause, until grading of these responses is received. Depending on the grading the process will either stop or continue. Last not least the Sliding Window Manager will request a change of ranking for those experts, who

System and Method for automatically managing response times in expertise systems based on an expert notification mechanism and - cc

1. replied in time
2. provided also a correct answer (based on the grading)

4. Expert Timer Manager

This component receives a small list of experts from the Sliding Window Manager initially. Gradually this component will receive more names of experts. In the best case, the number of concurrent experts to process will equal the "desired window size" w .

For each expert, the Expert Timer Manager will create and initialize a timer. This information is then sent to the Response Manager immediately, who will format the data, and notify the specified expert. There will be a timeout, if there's no response received from the specified expert via the Response Manager within a particular amount of time. The timeout can be specified from the originator of the problem/question, or a default value can be used. When a timeout occurs, the Expert Timer Manager will send a notification to the Sliding Window Manager, forwarding the name of the expert who didn't reply within the time frame. Also the Ranking Processor will be notified, so that this expert's ranking will decrease.

If a response is received from an expert after the timeout, the answer will not be discarded. It still could be useful. However, it will not help to increase the expert's response ranking.

If a response is received within the time frame, the Sliding Window Manager will also be informed about this.

5. Response Manager

The Response Manager will receive the name (email address, etc.) of an expert, along with the timing information, and a problem specification. It will format this data to the target platform (e.g. Email), and notify the expert. A reply form will also be included. The expert will have the possibility to decline a task, or to accept a task. In case of no reaction a timeout will occur. Acceptance can be defined if the expert replies with a possible solution. We might also implement a two level acceptance: First an expert has to formally accept, and then we would start an additional timer for the time the actual problem solving process. But these are implementation issues, which we won't discuss here.

In addition, this component has to automatically process the received answers. This could include parsing a received message, determine the expert, the problem solving task id, etc. It will only communicate with the Expert Timer Manager, which will take care of the coordination. To sum up, this component receives requests, processes replies, but has no knowledge about timing etc., which is handled on the layer above (Expert Timer Manager).

6. Ranking Processor

This component will receive ranking requests from the Expert Timer Manager or the Sliding Window Manager. Essentially the request could be to decrease or increase a ranking for responsiveness for a particular expert. For this reason the Ranking Processor will interact with the Expertise Ranking Database to update the data.

System and Method for automatically managing response times in expertise systems based on an expert notification mechanism and - cc



System Architecture.doc

[Figure 1: System Architecture Diagram with Flow Chart Information]



Expertise Presentation.p

[Figure 2: Disclosure Presentation]